UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/449,912	12/02/1999	NICK P. DIVITTORIO	202232	6873

7590 11/16/2007

LEYDIG VOIT & MAYER LTD
TWO PRUDENTIAL PLAZA SUITE 4900
180 NORTH STETSON
CHICAGO, IL 60601-6780

EXAMINER	
TANG, KENNETH	
ART UNIT	**PAPER NUMBER**
2195	

MAIL DATE	DELIVERY MODE
11/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/449,912
Filing Date: December 02, 1999
Appellant(s): DIVITTORIO, NICK P.

Divittorio
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 9/6/07 appealing from the Office action mailed 10/6/06.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,549,945	SINIBALDI	4-2003
5,526,794	MESSIH	6-1996
6,874,144	KUSH	4-1999
6,105,053	KIMMEL et al.	6-1995

- "BACKGROUND OF THE INVENTION"/Admitted Prior Art of Applicant's Specification, pages 1-3.

- Buchanan, Matt, et al. "Coordinated Thread Scheduling for Workstation Clusters Under Windows NT", Proceedings of the USENIX Windows NT Workshop, August 1997, pp. 3, 1st column, and page 5, Section 2.2.3.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. Claims 1-7, 13-19, and 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior art in the Specification (hereinafter AAPA) in view of Sinibaldi et al. (hereinafter Sinibaldi) (US 6,549,945 B1).

2. As to claim 1, AAPA teaches a control processor for executing a set of control tasks defining interactive control of an industrial process *(page 3, lines 1-2)*, the control processor comprising:

an embedded control task comprising a program including a set of output values corresponding to process setpoints *(page 2, lines 5-23)*;

a set of control blocks including regulatory control blocks having output values that are transmitted by the control processors to field devices coupled to the industrial process *(page 2, lines 5-23)*.

AAPA is silent in teaching a multi-variable linear program and having a high and low execution priority status wherein there is execution at a lower execution priority. However, Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). In addition, Sinibaldi teaches having high priority tasks such as foreground tasks as well as lower priority (non critical) tasks such as background tasks. Sinibaldi teaches executing in the lower priority by using background tasks (col. 17, lines 45-55). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the features of a matrix (multi-variable linear model) and priority execution because it would increase flexibility (col. 1, lines 50-55).

3. As to claims 2, AAPA teaches wherein the set of control blocks comprise supervisory control blocks *(page 2, lines 5-23)*.

4. As to claim 3, AAPA teaches wherein the supervisory control blocks include a multivariable control block including computer instructions facilitating communication between the control processor and a workstation *(see rejection of claims 1 and 2)*. In addition, Mann teaches downloading data between the control processor and device *(col. 13, lines 16-17)*.

5. As to claim 4, AAPA teaches wherein the multivariable control block includes a process control model to be implemented by the embedded control task *(see rejection of claim 1)*. Sinibaldi teaches downloading program instruction data between the control processor and device *(col. 19, lines 4-10, see Abstract)*.

6. As to claims 5, AAPA in view of Iino teaches wherein the supervisory control blocks include at least one multivariable loop block, and further comprising the step of execution of instructions and data associated with the at least one multivariable loop block *(see rejections of claims 1 and 2)*. AAPA teaches providing in put value for a regulatory control block via a user interface *(page 3, lines 1-2)*.

7. As to claim 6, AAPA teaches wherein regulatory control block is a PID block *(page 2, lines 5-23)*.

8. As to claims 7 AAPA is silent in teaching wherein the regulatory control block is a ratio block. However, it is well known in the art that control blocks can take on ratio values. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of the control block being a ratio block because this increases the functionality by being able to use ratio values as well as non-ratio values.

9. As to claims 13-19, they are rejected for the same reasons as stated in the rejections of claims 1-7, respectively.

10. As to claims 25, it is rejected for the same reasons as stated in the rejection of claim 1.

11. As to claims 26, it is rejected for the same reasons as stated in the rejection of claim 1. In addition, Sinibaldi teaches temporarily halting a background routine so that a foreground routine can be executed (col. 17, lines 45-55).

12. Claims 8-12 and 20-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior art in the Specification (hereinafter AAPA) in view of Sinibaldi et al. (hereinafter Sinibaldi) (US 6,549,945 B1), and further in view of Messih et al. (hereinafter Messih) (US 5,526,794).

13. As to claims 8-12, AAPA and Sinibaldi teach wherein the set of control blocks includes a supervisory control block including a sequence of instructions/tasks. They fail to explicitly teach a re-commencing cycle of the embedded task in accordance with a value specified by a repetition cycle parameter having a period, wherein the period specified by the repetition cycle parameter exceeds a period specified by the block processing cycle parameter. However, Messih teaches background and foreground execution in a controller wherein there is a time period (when period exceeds the period of completion of the foreground routine) before a cycle is restarted (col. 3, lines 20-34). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Messih to the existing system because this allows for

optimization of speed (increasing) and the necessary amount of time (decreasing) *(col. 4, lines 7-19)*.

14. As to claims 20-24, they are rejected for the same reasons as stated in the rejections of claims 8-12, respectively.

(10) Response to Argument

In page 6, last paragraph of the Appeal Brief, Applicant gives an overall argument that nowhere does the prior art provide a reason for implementing this specific multi-level control program execution scheme. Therefore, the claimed invention would not have been obvious to one of ordinary skilled in the art at the time of the invention.

In response, Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). This matrix is a multi-variable table which lists the tasks that are used during execution. The use of Sinibaldi's matrix reads on the limitation of the multi-variable linear program. AAPA (page 1, lines 12-34) and Sinibaldi (col. 2, lines 26-67) both relate to executing on a computer processor to control the operation of devices. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the features of a matrix (multi-variable linear model) and priority execution because it would increase flexibility in particularly in the execution of tasks (col. 1, lines 50-55). This is the general response to the general argument

made by the Applicant. The Examiner below has specifically addressed how the combination of AAPA and Sinibaldi teaches the claimed invention. See below.

Rejection of Claims 1-7, 13-19 and 25-26 as Obvious over AAPA in view of Sinibaldi

Claims 1, 6, 7, 13, 18, 19, 25, and 26

Applicant argues that there was no reason or teaching for the specific execution scheme recited in claim 1 wherein a multivariable linear program executes at a relatively low priority than a set of control blocks on a control processor for an industrial process control system. Applicant submits that there is an absence of any reason for one skilled in the art at the time of the invention to modify Applicant's Admitted Prior Art by incorporating the embedded control task comprising a multivariable linear program into a control processor, and executing the embedded control task at a lower priority than the set of control blocks.

In response, Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). This matrix is a multi-variable table which lists the tasks that are used during execution. The use of Sinibaldi's matrix reads on the limitation of the multi-variable linear program. AAPA (page 1, lines 12-34) and Sinibaldi (col. 2, lines 26-67) both relate to executing on a computer processor to control the operation of devices. Sinibaldi teaches having high priority tasks such as foreground tasks as well as lower priority (non critical) tasks such as background tasks. Sinibaldi teaches executing in the lower priority by using background tasks (col. 17, lines 45-55).

AAPA teaches and suggests the use of control blocks (page 2, 1-31) with supervisory logic using priority control (page 3, 9-11). Applicant admits on page 7, last paragraph of the Appeal Brief, that executing tasks based on priority is well known at the time of the invention. The Examiner agrees as class 718, subclass 103 is dedicated entirely to processing based on priority and operating systems that switch the task execution based on the priority/criticality of tasks have been around for decades. Therefore, the Supervisory program or control logic of AAPA would obviously use priority for coordinating its control of operation and processes since such techniques are well known in the art as illustrated by Sinibaldi (page 2, lines 13-23).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the features of a matrix (multi-variable linear model) and priority execution because it would increase flexibility in particularly in the execution of tasks (col. 1, lines 50-55). Flexibility/adaptability in task execution ensures non-critical lower priority tasks are not preventing the execution of critical, higher priority tasks. In addition, AAPA substantially teaches both components. AAPA teaches an embedded control task (Supervisory Program) that executes on the control processor, and control blocks (regulatory control blocks) that also executes and resides on the control processor (see pages 1-3, in particular, pg. 2, lines 5-23). The Supervisory Program coordinates the operation of the multiple control points and establishes setpoint values for the PID control blocks and regulatory blocks. The output values of supervisory control programs are used in the PID and other regulatory control blocks (Spec, page 2, lines 13-24). The regulatory control blocks are responsible for input and output signals received from and transmitted to field devices (see page 2, lines 5-12). Based on the teachings from AAPA, the only missing element would be that the embedded control task executes at a

lower execution priority than the set of control blocks. Applicant admits that priority scheduling/execution is well known in the art. The Examiner agrees and provides Sinibaldi as evidence of such. Typically, in order to execute one program out of a plurality of programs on a processor, that programs priority is increased, e.g. boosted, higher than the others. Therefore, since both the supervisory program and control blocks execute on the processor, their execution is done by the well known technique of boosting a program's priority for that task to execute. Hence, when it's the control blocks time to execute, its priority is boosted higher than the supervisory program such that the processor executes its functions. The references cited in the evidence section are examples of the well known teaching of priority scheduling, as detailed by Applicant as being well known in the art and an example of which is taught by Sinibaldi.

The Examiner furthermore states that motivation for an obviousness rejection is not solely based on it being explicitly disclosed by a reference. Motivation can be found based on knowledge generally available to one of ordinary skill in the art, MPEP 2143.01. Clearly as outlined above, a well known teaching for priority scheduling of programs on a processor would motivate one to perform the same priority scheduling on a control processor that executes a supervisory program and control blocks.

Applicant further argues that there is no reasonable expectation that some benefit would arise from the priority execution scheme detailed by AAPA in view of Sinibaldi. Applicant then states that the computational burden placed by the multivariable program on the control processor could limit the control processor's ability to perform desired functions, and risk that neither component will operate in a satisfactory manner thereby, if anything, reduce flexibility. The Examiner respectfully disagrees. The Examiner has outlined and further explained above

the motivation of combining AAPA and Sinibaldi. The Examiner has further explained the concept of priority scheduling/execution which boost the priority of programs such that higher priority programs are executed and no individual program exclusively controls the processor and how this technique applies to AAPA's use of a Supervisory Program and control blocks scheduled and executing on a control processor. The statements made by Applicant are not evidence to rebut this case of obviousness. In re Schulze, and MPEP 716.01(c) (II) details that arguments of counsel cannot take the place of evidence in the record, in particularly, an assertion of what seems to follow from common experience in just attorney argument and not the kind of factual evidence that is required to rebut a prima facie case of obviousness. Therefore, statements of what could limit a processor's ability or risk to run in an unsatisfactory manner are assertions by Applicant not capable of proof and cannot rebut the obviousness rejection.

Applicant argues that the contents of Sinibaldi's matrix/table are not generated from a multivariable linear program and Sinibladi does not teach that the multivariable linear program operates at a lower execution priority than an execution priority of a set of control blocks.

Applicant's arguments are repetitive. See above response by the Examiner. In addition, Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). This matrix is a multi-variable table which lists the tasks that are used during execution. The use of Sinibaldi's matrix reads on the limitation of the multi-variable linear program. AAPA and Sinibaldi both relate to executing

on a computer processor to control the operation of devices. Sinibaldi teaches having high priority tasks such as foreground tasks as well as lower priority (non critical) tasks such as background tasks. Sinibaldi teaches executing in the lower priority by using background tasks (col. 17, lines 45-55). AAPA teaches and suggests the use of control blocks (page 2, 1-31) with supervisory logic using priority control (page 3, 9-11). Applicant admits on page 7, last paragraph of the Appeal Brief, that executing tasks based on priority is well known at the time of the invention. The Examiner agrees as class 718, subclass 103 is dedicated entirely to processing based on priority. Making decisions on processing control is not novel and involves order and priority. Therefore, the Supervisory program or control logic of AAPA also uses priority for coordinating its control of operation and processes (page 2, lines 13-23).

Applicant argues that the motivation given in the Office Action is not proper because there is no reasonable expectation of success that some benefit would arise from executing a multivariable linear program on a control processor at a relatively lower priority than the control block execution task.

In response, this argument has already been addressed above. Furthermore, Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). This matrix is a multi-variable table that is used during execution. The use of Sinibaldi's matrix reads on the limitation of the multi-variable linear program. AAPA and Sinibaldi both relate to executing on a computer processor to control the operation of devices. Sinibaldi teaches having high priority tasks such as foreground tasks as well as lower priority (non critical) tasks such as background tasks.

Sinibaldi teaches executing in the lower priority by using background tasks (col. 17, lines 45-55). AAPA teaches and suggests the use of control blocks (page 2, 1-31) with supervisory logic using priority control (page 3, 9-11). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the features of a matrix (multi-variable linear model) and priority execution because it would increase flexibility (col. 1, lines 50-55). Flexibility/Adaptability of task priority allows for non critical tasks to be in the background while high priority tasks are in the foreground. This is desirable because the more important tasks will get executed before the less important tasks.

Applicant argues that the prior art does not disclose or provide a reason to modify AAPA such that the control processor executes the specifically recited control block/multivariable linear program execution scheme recited in claim 1.

Applicant continues to make the same argument. See the Examiner's response above. IN addition, Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). This matrix is a multi-variable table that is used during execution. The use of Sinibaldi's matrix reads on the limitation of the multi-variable linear program. AAPA and Sinibaldi both relate to executing on a computer processor to control the operation of devices. Sinibaldi teaches having high priority tasks such as foreground tasks as well as lower priority (non critical) tasks such as background tasks. Sinibaldi teaches executing in the lower priority by using background tasks (col. 17, lines 45-55). AAPA teaches and suggests the use of control blocks (page 2, 1-31) with supervisory

logic using priority control (page 3, 9-11). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the features of a matrix (multi-variable linear model) and priority execution because it would increase flexibility in particularly in the execution of tasks (col. 1, lines 50-55). Flexibility/adaptability in task execution ensures non-critical lower priority tasks are not preventing the execution of critical, higher priority tasks.

Applicant admits in the Remarks that executing various program tasks at different priority levels was indeed well known at the time of the invention. However, Applicant argues that there is no teaching or suggestion for one of ordinary skill in the art at the time the application was filed to incorporate the embedded multivariable linear program that calculates set point values for the control processor into the process controller and execute the multivariable application program at a lower assigned priority than a set of control blocks.

Firstly, it is not claimed that there is calculation of set point values. In response, Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). This matrix is a multi-variable table that is used during execution. The use of Sinibaldi's matrix reads on the limitation of the multi-variable linear program. AAPA and Sinibaldi both relate to executing on a computer processor to control the operation of devices. Sinibaldi teaches having high priority tasks such as foreground tasks as well as lower priority (non critical) tasks such as background tasks. Sinibaldi teaches executing in the lower priority by using background tasks (col. 17, lines 45-55). AAPA teaches and suggests the use of control blocks (page 2, 1-31) with supervisory logic using

priority control (page 3, 9-11). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the features of a matrix (multi-variable linear model) and priority execution because it would increase flexibility in particularly in the execution of tasks (col. 1, lines 50-55). Flexibility/adaptability in task execution ensures non-critical lower priority tasks are not preventing the execution of critical, higher priority tasks. Applicant continues to make the same arguments. See Examiner's response above.

Claims 2-3, 5, 14, and 17

Applicant argues that there is no teaching to execute in the multi-level execution scheme.

See response to arguments above. In addition, a "Multi-level" is not claimed but rather multi-variable. Sinibaldi teaches having a matrix which represents a multi-variable linear program, wherein the entries of the matrix are setpoints (col. 18, lines 25-28, Fig. 14). This matrix is a multi-variable table that is used during execution. The use of Sinibaldi's matrix reads on the limitation of the multi-variable linear program.

Claims 4 and 16

Applicant argues that the prior art does not even remotely teach the use of a multivariable control block to download a process control model for implementation by the embedded control task.

In claims 4 and 16, "downloading" is not claimed but rather "receiving and storing", which is slightly different in scope. Nevertheless, AAPA teaches transmitting control signals to

controlled elements of the industrial process and for receiving (downloading) industrial process status data (see page 4, lines 5-9). Sinibaldi teaches downloading program instruction data between the control processor and device *(col. 19, lines 4-10, see Abstract)*.

Claim 15

Applicant argues that the prior art does not even remotely suggest the recited step of downloading data from a workstation to a database accessed by a multivariable control block.

Sinibaldi teaches transferring data between a host network 5, such as a database service network, and the computer system1 over a Local Area Network 6, or other connection (col. 2, lines 66-67 through col. 3, lines 1-2). AAPA teaches computer systems having a multivariable control block *(page 2, lines 5-23)*

Rejection of Claims 8-12 and 20-24 as Obvious over AAPA in view of Sinibaldi and Messih

Applicant argues that the Office Action does not explain how the teachings of Messih apply to the specific multi-level control program execution scheme (including an embedded task and a set of control blocks operating at different priorities) recited in each of the claims.

15. Messih teaches background and foreground execution in a controller wherein there is a time period (when period exceeds the period of completion of the foreground routine) before a cycle is restarted (col. 3, lines 20-34). One of ordinary skill in the art would have been able to

take the same rationale of background/foreground execution to apply it to the control system of Sinibaldi and AAP.

Claims 8, 11, 20, and 23

Applicant argues that the Office Action does not identify the repetition cycle parameter.

Messih teaches background and foreground execution in a controller wherein there is a time period (when period exceeds the period of completion of the foreground routine) before a cycle is restarted (col. 3, lines 20-34). AAPA also discloses the supervisory control program responding faster to changes in dynamic parameters of the controlled process (page 3, lines 8-11).

Claims 9 and 21

Applicant argues that the prior art does not disclose or suggest the claimed supervisory control block, within the set of control blocks, that controls commencing a repetition cycle of the embedded task.

Messih teaches background and foreground execution in a controller wherein there is a time period (when period exceeds the period of completion of the foreground routine) before a cycle is restarted (col. 3, lines 20-34). AAPA also discloses the supervisory control program responding faster to changes in dynamic parameters of the controlled process (page 3, lines 8-

11). The supervisory control program that associates with the control blocks are supervisory control blocks (see AAPA, page 2, lines 5-31).

Claims 12 and 24

Applicant argues that the references do not disclose or suggest embedded repetition period exceeding a block processing cycle period.

Messih teaches background and foreground execution in a controller wherein there is a time period (when period exceeds the period of completion of the foreground routine) before a cycle is restarted (col. 3, lines 20-34). When the time period is exceeded or expired, then the cycle is restarted.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Kenneth Tang

Conferees:

MENGAL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER